Ultimate Tic-Tac-Toe AI

This project will require you to build an Ultimate Tic-Tac-Toe game, where a user can play against an AI agent.

In class we discussed adversarial games, with Tic-Tac-Toe as an example using MiniMax search. In this assignment, extend what was done in class (or write from scratch) a program to play Ultimate Tic-Tac-Toe against a human player.

Ultimate Tic-Tac-Toe is similar to regular Tic-Tac-Toe except we play mini games inside each square. See http://mathwithbaddrawings.com/2013/06/16/ultimate-tic-tac-toe/ for a detailed description of the game-play. Use this site to try the game: http://ultimatetictactoe.creativitygames.net/

There are **2 rule modifications:**

1. If you or the AI agent is sent to a board that is already won by any player or tied (no more empty squares to place your piece in), you are to play on the *first* playable board from the beginning. Left to right, top down. If no boards are playable, then the result is a tie. This should vastly simplify your game/AI play.
2. If a small board is tied, neither player can count it as a win in the overall game.

Implement any AI agent strategy:

1. MiniMax depth limited.
2. Alpha-Beta pruning MiniMax depth limited.
3. Monte Carlo game search.
4. For any other strategy, please ask before starting.

Your Agent should compute its next move within 30 seconds of its turn.

Use the text-based interface developed in the Tic-Tac-Toe code from class. The game should be easy to use for someone who knows how Ultimate Tic-Tac-Toe works. Be sure to print out how to enter moves.

Your grade will consist of the following:

1. 20% - Submitting code.
2. 40% - Playing the game is possible, but the AI agent is easily beatable. Correctly implements the game-play. (AI not operating)
3. 20% - AI agent is not easily beatable, but takes longer than 30 seconds to deliberate per turn.
4. 20% - AI agent is not easily beatable, and takes less than 30 seconds to deliberate per turn.